

## THE ADVANCED SOFTWARE DEVELOPMENT WORKSTATION PROJECT

Ernest M. Fridge III and Charles L. Pitman  
Software Technology Branch  
NASA/JSC/PT4, Houston, TX 77058

## ABSTRACT

The Advanced Software Development Workstation (ASDW) task is researching and developing the technologies required to support Computer Aided Software Engineering (CASE) with the emphasis on those advanced methods, tools, and processes that will be of benefit to support all of the National Aeronautics and Space Administration (NASA). Immediate goals are to provide research and prototype tools that will increase productivity, in the near term, in projects such as the Software Support Environment (SSE), the Space Station Control Center (SSCC), and the Flight Analysis and Design System (FADS) which will be used to support the Space Shuttle and the Space Station Freedom. Goals also include providing technology for future SSE and operational systems by adding knowledge based system support to all phases of information systems development, evolution, maintenance, and operation. The technologies under research and development in the ASDW project are targeted to provide productivity enhancements during the software life cycle phases of enterprise and information system modeling, requirements generation and analysis, system design and coding, and system use and maintenance. A programmable, Zachman-style framework is planned that will guide the information system modeling process and will be supported by system modeling tools integrated by a common knowledge base. An engineering graphical language will permit engineers to design applications and application templates. A software parts composition system will provide the environment for accessing parts, for "filling in the blanks" in generic parts, and for assembling parts based on the application templates. On-line user's guides will assist users in operating the developed information system with knowledge base expert assistance.

## INTRODUCTION

Software development is a serious bottleneck in the construction of complex information systems, during both the development and evolution of such systems (Figure 1). Both development costs and maintenance costs can be high. The heaviest development costs tend to occur in the early part of the total life cycle during requirements generation, requirements analysis, design, and application development. Maintenance costs for sustaining the developed information system are even higher. An increase of the reuse of any of the software "parts" used in these activities has been viewed as a way to relieve this bottleneck. One approach to achieving software reusability is through the development and use of software parts composition systems [1,2].

A software parts composition system is a software development environment comprised of a parts description language for modeling parts and their interfaces, a catalog of existing parts, a composition editor that aids a user in the

specification of a new application from existing parts, and a code generator that takes a specification and generates an implementation of a new application in a target language.

The Advanced Software Development Workstation (ASDW) is currently an expert system shell that provides the capabilities required to develop and manipulate these software parts composition systems. The ASDW is now in Beta testing at the Johnson Space Center. Future work centers on responding to user feedback for capability and usability enhancement, expanding the scope of the support for collecting, representing, and manipulating knowledge during the early phases of the information system life cycle (Figure 1), and in providing solutions for handling very large libraries of reusable components.

## APPROACH

The ASDW is now moving into phase IV which will significantly broaden its scope of influence. Phase I (October, 1985 to April, 1987)

demonstrated the feasibility of a knowledge based approach to application generation in a limited domain. Phase II (April, 1987 to February, 1989) investigated ways to exploit the use of knowledge representation, retrieval, and acquisition techniques. A prototype demonstrated a knowledge based system for the development of software parts composition systems (i.e., a software parts composition shell). Phase III (March, 1989 to December, 1989) prototyped ways to handle the scale-up problem (1000-100000 objects), prototyped ways to automatically generate the taxonomy, and initiated two Beta test projects at JSC. One test project is to generate trajectory mission planning simulations from software parts and the other is to provide expert assistance to operational users in setting up input data for simulations. The projects will support the SSCC and FADS, respectively.

During Phases II and III, the project also began joint activity with the United States Air Force (USAF), studying the information requirements of information systems, their integration and development processes, the methodologies required, and the integrated tools and integrated knowledge base that supports this development. The USAF is providing funding annually to study the modeling, methodologies, and information requirements of manufacturing information systems [3]. Modeling is based on the USAF's IDEF methodologies; IDEF is an acronym for ICAM (Integrated Computer Aided Manufacturing) Definition. JSC added funds to expedite the development of two methodology tools and a computer-assisted tutor to educate developers in the proper use of the methodologies.

Today, ASDW is the basic shell for supporting the reuse of stored information whether it be about software artifacts or any other design artifact. It contains object management and rule based constraint handling as well as a sophisticated "point and click" textual user interface (called "Specification-by-Reformulation") that models the way that people communicate among themselves. A neural net approach has been implemented to handle the large number of information objects that can be stored and a capability to automatically generate the taxonomy of objects has been incorporated. The "Help" system uses hypermedia technology.

Field testing by users in JSC's mission planning community was initiated in fiscal year 1989 (FY89) and has been a major thrust in FY90. Also, the user interface windowing system was made more portable with the migration to X-windows and TAE Plus [4]. In FY91, the total number of objects will be increased to handle a volume in the neighborhood of 100,000, and the user interface will be made more graphical with the capability to directly define application templates from a block diagram point of view. The modeling, generation, and reuse of early life cycle artifacts will be added along with the capability to use integrated methodologies and a common representation of their information content such that they can share common information.

In FY91, a programmable, Zachman-style framework [5] that will guide the knowledge acquisition and modeling process, the proper use of information system modeling tools, and the selection of the proper methodologies to be used, as a function of the characteristics of the information system, will be developed. This framework will be supported with an "Integrated Platform" of integrated services and a knowledge representation language that will integrate modeling tools used to define the information systems. These modeling tools will enforce the correct use of the methodologies.

Also during FY91, field testing will define techniques to support users in acquiring the knowledge required to operate the developed information system and will provide capability to assist users in the set up of operational data input with knowledge base expert assistance.

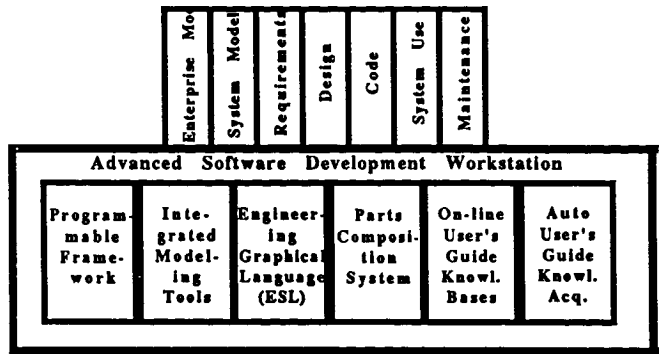
During Phase IV (which began January, 1990), CASE research and development will continue with the support of four organizations: Inference Corporation, SofTech, the Knowledge Based Systems Laboratory (KBSL) at Texas A&M University, and the National Research Council (NRC).

Inference will provide the reusable software library management system, and the user interface for accessing parts, for specifying new parts, and for assembling them to create applications. These capabilities will be incorporated into the component called Bauhaus [1,2].

SofTech will define an engineering graphical language called Engineering Script Language (ESL) which should significantly increase productivity in application generation. The ESL is a high level graphical language that permits engineers with a minimum of programming training to design applications that are populated with parts from the Bauhaus library. Constraint checking of the graph and the parts selection will be provided. The ESL is based upon proven concepts which have been put into operation at the Naval Research Laboratory in a restricted domain [6]. The ASDW effort will demonstrate its utility in the mission planning and analysis domain. The ESL will be integrated with the knowledge based library of reusable parts.

The KBSL will develop a programmable, Zachman-style framework [5] of information systems requirements that will aid in the use of methodologies (based on extended IDEF methodologies), integrated methodology tools, and the theory of information modeling (what models are required, when and how to produce them, how to use them, how to integrate their knowledge, etc). It will also aid in determining the activities for the various developer and user roles across the life cycle. All of the activities in the framework may be configured to match the characteristics of the information system being supported (e.g., business vs engineering). This on-line framework will be supported by and will help manage a prototype "Integrated Platform" which provides a uniform knowledge representation that will integrate methodologies and the models produced by them. This uniform representation of knowledge is the basis for integration and reuse of information among activities and information products throughout the life cycle of the information system.

Finally, the NRC will provide research into the use of expert systems to support the operational environment of the information system [7]. This study will develop a method to create a knowledge base, while an application is being developed, which can then help users easily set up complex input data in the completed application.



**Figure 1 ASDW Project**

The ASDW project consists of six research and development efforts aimed at supporting Computer Aided Software Engineering (CASE) throughout all phases of the software life cycle.

## CONCLUSION

The CASE research and development being performed by the ASDW task is general in its scope and should benefit many NASA programs. The following sample benefits are anticipated for system developers, operators, and maintainers.

For information system developers, there should be an increase in productivity by providing application generation, parts composition, and an engineering graphical language. Reuse of all types of system development artifacts will be enhanced, and the knowledge existing in the various models of a system's phases will be integrated and translated through the life cycle. A goal is to provide integration of system modeling tools and advice on when and how to use the tools as a function of the type of information system to be developed; in particular, a goal is to provide an integrated knowledge base of the system's development which assists and guides the entire system development process.

For information system operators, the development of a prototype tool to create and maintain intelligent interfaces should increase productivity by providing on-line user's guides that include a hypertext help facility and a knowledge base to support data selection, to test for constraint violations, to generate input data streams and command streams from templates,

and ultimately to supply expert advice on system operation, as a function of the operator's level of expertise.

For information system maintainers, all of the benefits to system developers will be available. Additionally, the integrated knowledge base built up automatically during an information system's development will capture the appropriate knowledge of the system's original developers and make this available to the system maintainers.

The ASDW project is currently field testing and refining the parts composition system and the on-line user's guide builder. Work is continuing on the engineering graphical language, the application template generation, the programmable framework, the integrated system modeling tools, and the integrated system knowledge base.

## REFERENCES

1. Allen, Bradley and Lee, Daniel, "A Knowledge-based Environment For the Development Of Software Parts Composition Systems," ACM 0270-5257/89/0500/0104, pp. 104-112, 11th International Conference On Software Engineering, Pittsburgh, PA, May 15-18, 1989.
2. Allen, Bradley, "Case-based Reasoning In a Software Information System," submitted to AAAI-90/National Conference On Artificial Intelligence, Boston, MA, July 29 - Aug. 3, 1990.
3. Mayer, Richard, et. al., "Development Methodologies For Integrated Information Management Systems: Final Technical Report," Air Force Human Resources Laboratory/LRA, WPAFB, OH, July, 1988.
4. NASA Goddard Space Flight Center, "Transportable Applications Environment Plus," GSC-13275 with documentation, Computer Software Management and Information Center (COSMIC) at the University of Georgia, Athens, GA, Jan., 1990.
5. Zachman, John, "A Framework For Information Systems Architecture," G320-2785, IBM, Los Angeles, CA, March, 1986.
6. SofTech, Inc., "Graph Analysis And Design Technique Methodology," SofTech, Inc., Falls Church, VA, Nov., 1984.
7. Izygon, Michel, "Intelligent Interface For Complex Simulation Codes," Research Proposal to NASA/Johnson Space Center, National Research Council: Research Associateship Program, Washington, D.C., Sept., 1989.